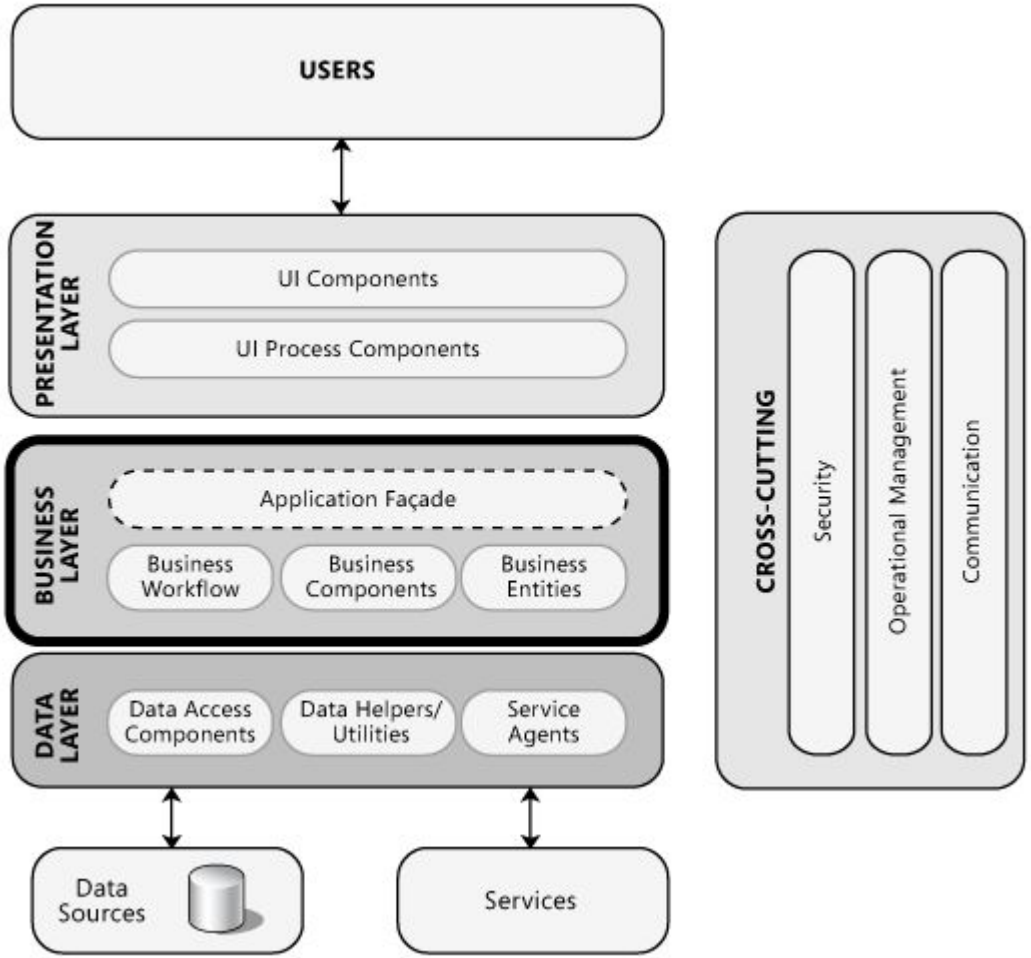
Security patterns

Agáta Dařbujanová
Petr Domkař



Zdroj: https://msdn.microsoft.com/en-us/library/ee658103.aspx

Autentizace

K obraně před útoky na aplikaci a odcizení identity je vhodné se řídit následujícími principy:

- Určit hranice zabezpečení aplikace
- Ověření uživatele na klientovi i serveru
- Vynutit bezpečné heslo
- Hesla ukládat a posílat v zabezpečeném formátu
- Má-li uživatel přístup k více aplikacím se stejným ověřením - vhodné použít Single sign-on Strategy

Single sign-on strategy [5]

- Služba zajišťující přístupová práva díky jednomu ověření pro celý běh ve více aplikacích.
- Zajišťuje samostatný model aplikace
- Možné využití protokolu Kerberos a SAML
- Pohodlné pro uživatele
- Při napadení útočníkem přístup ke všem aplikacím
  - Každý aspekt - ověřovat totožnost
- Možné využít dvoufázové autentizace (2FA)

Autorizace

Nutné ověření k čemu má ověřený uživatel přístup. Dobré zvážit

- Oprávnění na rolích
  - Uživatelé rozděleni do rolí - lehčí správa oprávnění
- Oprávnění na zdrojích
  - Určit dané zdroje - pomocí ACL přiřadit práva uživatelům (Rolím)
- Oprávnění na nárocích
  - Usnadňuje oddělení pravomocí autorizace od autorizačního a autentizačního mechanismu.

Bezpečnostní doporučení pro business logiku [1]

- zranitelnosti business logiky jsou jiné, nelze je snadno klasifikovat (př. editace příspěvku)
    1. identifikace business pravidel a testování
        a. pravidlo bylo implementováno
        b. je prováděno správně a nelze obejít
        c. je použito všude tam, kde má být

zvážení business pravidel souvisejících s:

2. časem
3. financemi
4. procesem
5. lidskými zdroji
6. smlouvou

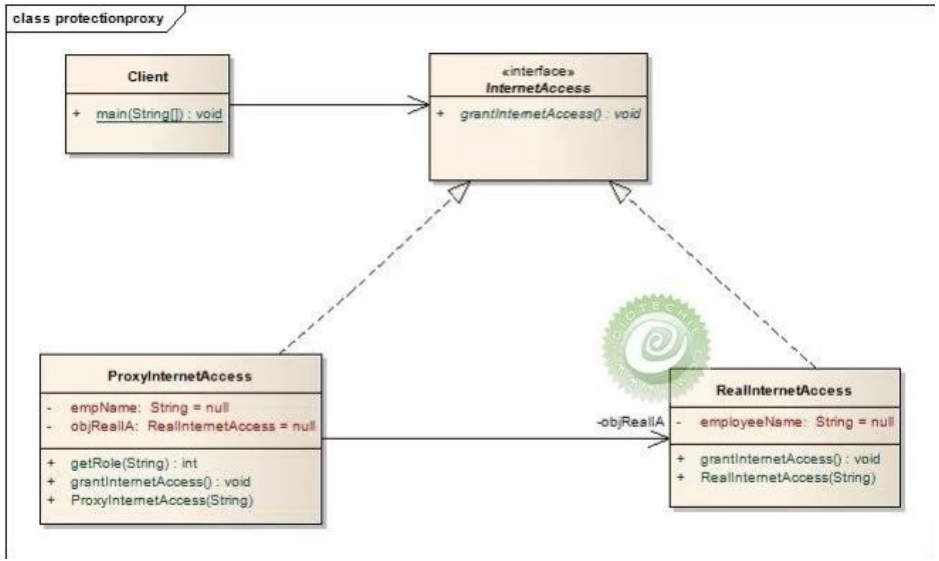
Security layer

- implementováno 2x [2]:
  - business vrstva -- chrání uživatelova data
  - view vrstva -- uživatel vidí pouze pro něj relevantní data
- zajistit vše v 1 vrstvě
- `UI -> Security -> BLL -> DAL (SOA → security service)`[3]
  - snazší údržba pravidel
- Decorator pattern [3]
  - návrhový vzor
  - přidání individuálního objektu bez vlivu na chování objektů ve stejné třídě
  - single responsibility principle
- kód je přehlednější a je snazší jej udržovat [4]

Protected system patterns [6]

Secure Proxy pattern

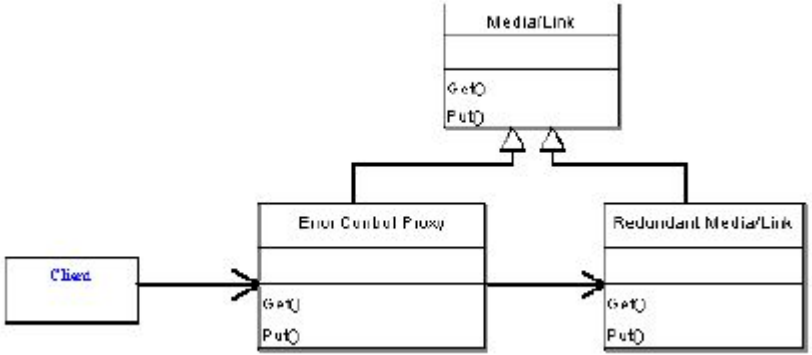
- Vhodné použít pro obalení hlavního objektu pro konkrétního uživatele
- Ve firmě nová politika s omezeným přístupem k internetu
  - RealInternetAccess poskytuje všem internet
  - ProxyInternetAccess
    - implementuje stejné rozhraní
    - poskytuje internet na základě rolí uživatelů



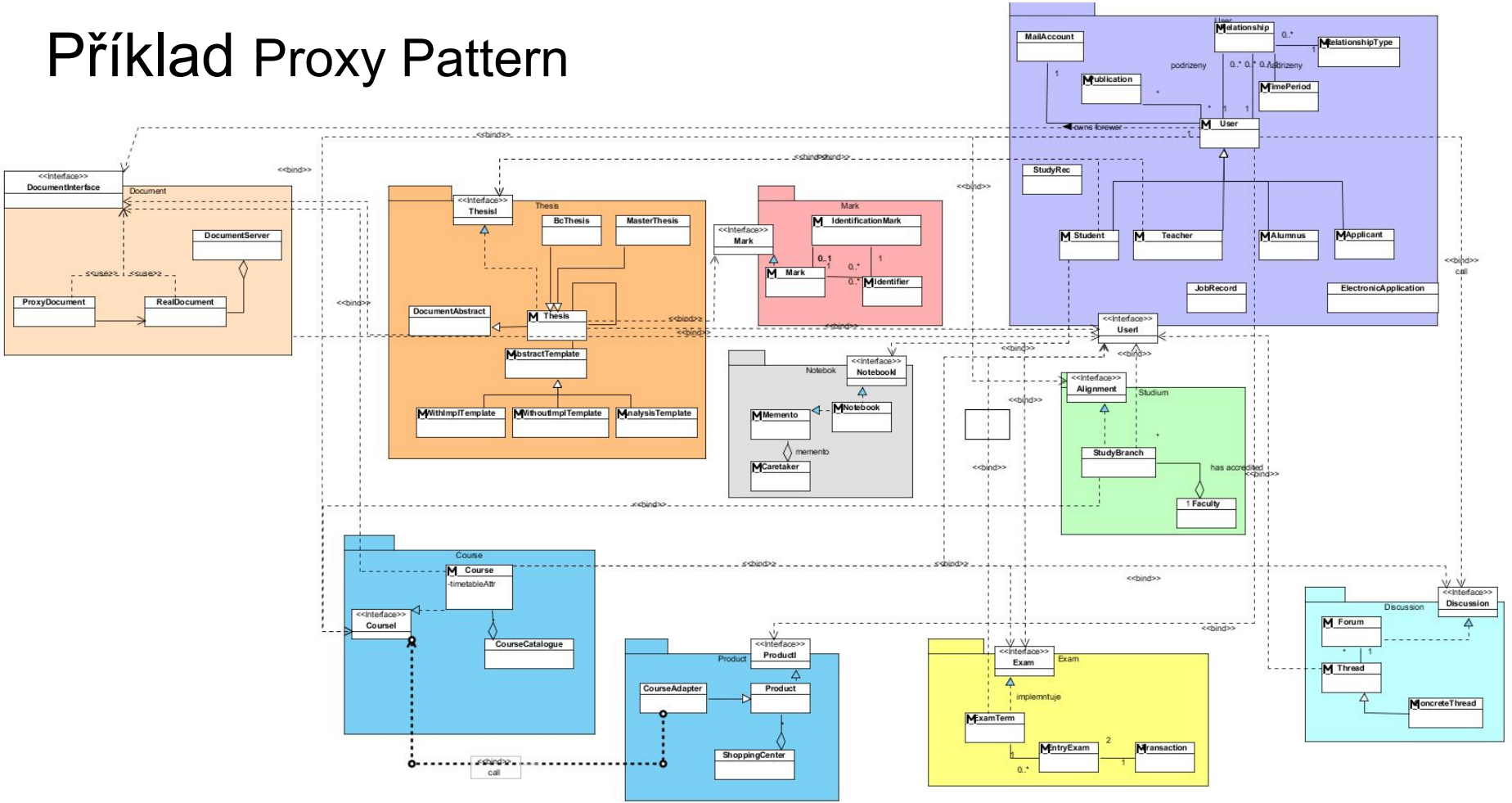
Available system patterns [6]

Detection/Correction pattern

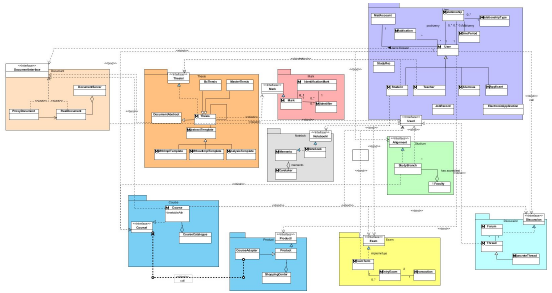
- přidává redundanci dat → detekce a obnova chyb

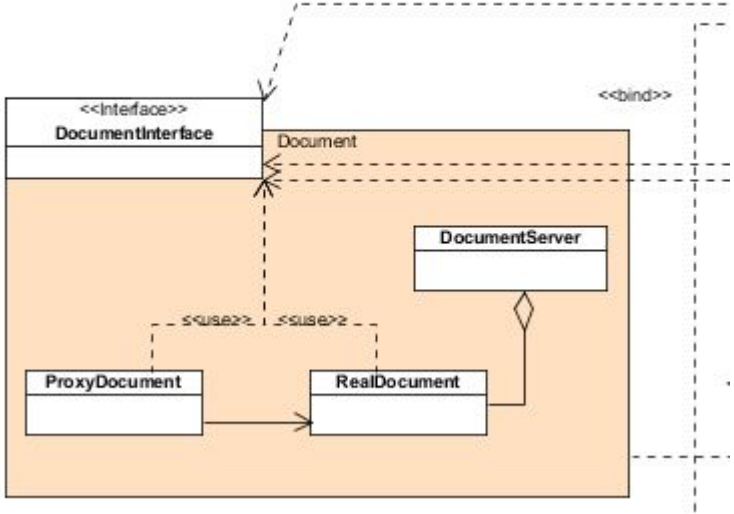


Příklad Proxy Pattern



Příklad Secure Proxy pattern





Děkujeme za pozornost

Zdroje

[1]https://www.owasp.org/index.php/Business_Logic_Security_Cheat_Sheet

[2]http://dontpanic.42.nl/2011/05/introducing-security-layer-in-your.html

[3]http://stackoverflow.com/questions/3007337/make-a-method-of-the-business-layer-secure-best-practice-best-pattern

[4]http://searchsecurity.techtarget.com/definition/single-sign-on

[5]http://idiotechie.com/gang-of-four-proxy-design-pattern/

[6]https://www.researchgate.net/publication/27383036_A_Qualitative_Evaluation_of_Security_Patterns